UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,935	05/23/2001	Sameer Siddiqui	NTECP001	1680

7590 08/24/2004

Sameer Siddiqui
1762 Technology Drive, Suite 226
San Jose, CA 95110

EXAMINER	
EDELMAN, BRADLEY E	

ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 08/24/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/864,935	SIDDIQUI, SAMEER	
	Examiner	Art Unit	
	Bradley Edelman	2153	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 May 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *23 May 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *8/29/01, 10/09/01*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

This is a first Office action on the merits of this application. Claims 1-20 are presented for examination.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

1. Claim 2 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The term "substantially" in claim 2 is a relative term which renders the claim indefinite. The term "substantially" is not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention.

2. Claims 1-12 and 18-19 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential steps, such omission amounting to a gap between the steps. See MPEP § 2172.01.

Claim 1 describes in the preamble "a method of sending network device instructions to a network device." However, the body of the claim does not describe either a "network device instruction" or a "network device." Thus, the claim lacks the

essential steps linking the network device mentioned in the preamble with the steps claimed in the body of the claims.

In further discussing claim 1, the claim also lacks essential steps of deriving proxy agent instructions from the application instructions, which would be necessary to carry out the step of "uploading proxy agent instructions derived from the application instructions to a mail server," and of downloading the proxy agent instructions by a proxy agent, which would be necessary to carry out the step of "confirming that the proxy agent instructions have been downloaded by a proxy agent." The claim is confusing without these essential steps, and the invention as claimed would not operate without these steps.

Claims 2-12 and 18 depend from claim 1 and are therefore rejected as well.

Claim 19 suffers from the same shortcomings as claim 1, and is thus rejected for the same reasons.

In a similar manner, claim 5 describes "determining the proxy agent that is associated with the network device," but does not describe associating a proxy agent with a network device.

Furthermore, claim 6 describes, "specifying the address of the proxy agent that is to execute the proxy agent instructions," but does not describe selecting or choosing a proxy agent is to execute the proxy agent instructions.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-11 and 13-20 are rejected under 35 U.S.C. 102(e) as being anticipated by L'Heureux et al. (U.S. Patent No. 6,697,942, hereinafter "L'Heureux").

In considering claim 1, L'Heureux discloses a method of sending network device instructions ("commands") to a network device ("DET") comprising:

Receiving an application instruction generated by an application (col. 5, lines 40-42, 55-59, wherein "GUI software 20" is used to "construct a complex message," which is received by a "command formatter 220");

Uploading proxy agent instructions derived from the application instructions to a mail server (col. 5, lines 55-67, col. 6, lines 1-5, wherein the "command formatter 220" converts the application instructions into "multiple data-type commands" and those commands are uploaded to the "SMTP server"); and

Confirming that the proxy agent instructions have been downloaded by a proxy agent (col. 6, lines 14-23, 29; col. 8, lines 48-50, wherein the e-mail is "downloaded" to the in-box at the DET, and the DET "automatically generates and sends a return e-mail message confirming at the embedded DET commands have been executed properly").

In considering claim 2, Examiner has interpreted the term "substantially the same" as simply meaning "the same." Thus interpreted, L'Heureux further discloses that the proxy agent instructions are the same as the application instructions (both are instructing the DET to perform the command).

In considering claim 3, L'Heureux further discloses encrypting the proxy agent instructions (col. 7, lines 60-63, wherein a "security key block" is used to encrypt the message).

In considering claim 4, L'Heureux further discloses authenticating the proxy agent instructions (col. 8, lines 4-5, "the decryption module 322 validates the message").

In considering claim 5, L'Heureux further discloses determining a proxy agent that is associated with the network device (col. 6, lines 1-11, wherein the POP server is the proxy agent, and the system determines the POP server associated with the device).

In considering claim 6, L'Heureux further discloses specifying the address of the proxy agent that is to execute the proxy agent instructions (col. 6, lines 60-65, wherein the in-box is the proxy agent and the sender specifies the address of the user's inbox).

In considering claim 7, L'Heureux, further discloses that confirming that the instructions have been downloaded by a proxy agent includes receiving an acknowledgment from the proxy agent that the instructions have been received (col. 8, lines 47-50, wherein the proxy agent is the user's e-mail program and it sends a confirmation message to the sender that the "DET commands have been executed properly").

In considering claim 8, L'Heureux further discloses that the mail server sends the proxy agent instructions to a proxy agent mail server from which the proxy agent may download the instructions (col. 6, lines 4-5, 13-14, 20-22, "SMTP server 130 then transfers the e-mail message to the target recipient's POP server 160," and "the e-mail message will be downloaded in the customary manner").

In considering claim 9, L'Heureux further discloses that the proxy agent downloads the instructions directly from the mail server (col. 5, lines 5-6, "the SMTP server 130 and the POP server 160 may be co-located" such that the messages are downloaded directly from the SMTP server).

In considering claim 10, L'Heureux further discloses that the proxy agent instructions are associated with a session identifier (col. 9, lines 4-10, wherein the "Data type x-clipmail identifies an e-mail message segment containing DET commands").

In considering claim 11, L'Heureux further discloses that the proxy agent instructions are associated with a session identifier and wherein the session identifier is used to identify the proxy agent instructions downloaded by the proxy agent (col. 9, lines 4-10, wherein the "Data type x-clipmail identifies an e-mail message segment containing DET commands").

In considering claim 13, L'Heureux discloses a method of receiving network device instructions ("commands") for a network device ("DET") comprising:

Downloading a message from a mail server (col. 6, lines 12-14, "the next time the target recipient logs into the POP server 160 for an e-mail session, the e-mail message will be downloaded"), the message including the network device instructions (col. 5, lines 55-67, "command");

Authenticating the message (col. 8, lines 4-5, "the decryption module 322 validates the message"); and

Parsing the instruction (col. 8, lines 5-6, "the parser module 312 separates the message into command data blocks").

In considering claim 14, L'Heureux further discloses that the network device instructions are derived from instructions generated by an application (col. 5, lines 40-41, 55-60, "GUI software").

In considering claim 15, L'Heureux further discloses sending an acknowledgment that the instructions have been received (col. 8, lines 47-50, "return e-mail message confirming that the embedded DET commands have been executed properly," and thus implicitly confirming that the instructions have been received).

In considering claim 16, L'Heureux further discloses executing the instructions and uploading the results to the mail server (col. 8, lines 47-50, "generates and sends return e-mail message confirming that the embedded DET commands have been executed properly").

In considering claim 17, L'Heureux further discloses managing a network according to the network device instructions (col. 4, lines 10-15; col. 5, lines 33-46, wherein configuring the remote computer constitutes managing the network).

In considering claim 18, L'Heureux further discloses gathering network data according to the network device instructions and reporting the results (col. 8, lines 47-50, wherein the network device "generates and sends return e-mail message confirming that the embedded DET commands have been executed properly" thereby gathering data regarding the results of the command and reporting the results to the sender).

In considering claim 19, L'Heureux discloses a proxy agent manager ("PC") for sending network device instructions ("commands") to a network device ("DET") comprising:

An application interface ("command formatter 220") configured to receive an application instruction ("command") generated by an application ("GUI," col. 5, lines 25-30, 40-41, 55-60); and

A mail server interface ("SMTP server 130") configured to upload proxy agent instructions derived from the application instructions to a mail server and configured to confirm that the proxy agent instructions have been downloaded by a proxy agent (col. 6, lines 1-5, 20-22, 29; col. 8, lines 47-50, wherein the server uploads the instructions and wherein the proxy agent at the DET responds with a confirmation through the same e-mail servers).

In considering claim 20, L'Heureux discloses a network device for executing instructions from a remote manager comprising:

A mail server interface configured to download a message from a mail server (col. 6, lines 12-14, "the next time the target recipient logs into the POP server 160 for an e-mail session, the e-mail message will be downloaded"), the message including the instructions (col. 5, lines 55-67, "command"); and

A processor configured to authenticate the message and to parse the instructions (col. 8, lines 4-6, "the decryption module 322 validates the message, the parser module 312 separates the message into command data blocks...").

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over L'Heureux, in view of features that are well known in the art.

In considering claim 12, although L'Heureux discloses using decryption and security keys to encrypt and secure messages sent across the network, L'Heureux does not disclose the use of a firewall, such that the network device is behind a firewall as claimed. Nonetheless, Examiner takes Official notice that the use of Firewalls on the Internet and over e-mail systems is well known in the art (for instance, the USPTO e-mail system has used firewalls for security since at least 1999). Given this knowledge, a person having ordinary skill in the art would have readily recognized the desirability and advantages of placing the network device taught by L'Heureux behind a firewall to increase security and prevent hackers from accessing the device. Therefore, it would have been obvious to place the DET taught by L'Heureux behind a firewall.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Bradley Edelman whose telephone number is 703-306-3041. The examiner can normally be reached from 9 a.m. to 5 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glen Burgess can be reached on 703-305-4792. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Bradley Edelman

BE
August 20, 2004

Form 1449 (Modified)	Atty Docket No.	NTECP001
Information Disclosure Statement By Applicant	Application No.:	09/864,935
	Inventor	SAMEER SIDDIQUI
	Group	2183
(Use Several Sheets if Necessary)	Filing Date	May 23, 2001

U.S. Patent Documents

Examiner Initial	No.	Patent No.	Date	Patentee	Class	Sub-Class	Filing Date
	A	5,968,116	Oct. 19, 1999	Day, II et al	709	202	Aug. 22, 1997
	B	5,968,124	Oct. 19, 1999	Takahashi, et al	709	224	Nov. 6, 1996
	C	6,125,390	Sept. 26, 2000	Touboul	709	223	Aug. 25, 1997
	D	6,145,001	Nov. 7, 2000	Scholl, et al	709	223	Jan. 5, 1998
	E	6,219,708	Apr 17, 2001	Martenson	709	226	May 30, 1996
	F	6,226,666	May 1, 2001	Chang, et al	709	202	June 27, 1997
	G						
	H						
	I						
	J						
	K						

Foreign Patent or Published Foreign Patent Application

Examiner Initial	No.	Document No.	Publication Date	Country or Patent Office	Class	Sub-class	Translation Yes	Translation No
	L							
	M							
	N							
	O							
	P							

Other Documents

Examiner Initial	No.	Author, Title, Date, Place (e.g. Journal) of Publication
	R	
	S	
	T	

Examiner	Bradley Edelman	Date Considered	8/20/04

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Form 1449 (Modified)	Atty Docket No.	NTECP001
Information Disclosure Statement By Applicant	Application No.:	09/864,935
	Inventor	Sameer Siddidqui
	Group	2183
(Use Several Sheets if Necessary)	Filing Date	May 23, 2001

U.S. Patent Documents

Examiner Initial	No.	Patent No.	Date	Patentee	Class	Sub-Class	Filing Date
	A						
	B						
	C						
	D						
	E						
	F						
	G						
	H						
	I						
	J						
	K						

Foreign Patent or Published Foreign Patent Application

Examiner Initial	No.	Document No.	Publication Date	Country or Patent Office	Class	Sub-class	Translation Yes	Translation No
	L							
	M							
	N							
	O							
	P							

Other Documents

Examiner Initial	No.	Author, Title, Date, Place (e.g. Journal) of Publication
	R	Deri, Luca, "Java-Based Mobile Asset Location", IEEE Monet Special Issue on Mobility, 1999, page1-13.
	S	
	T	

Examiner	Date Considered
Bradley Edelman	8/20/04

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Notice of References Cited	Application/Control No. 09/864,935	Applicant(s)/Patent Under Reexamination SIDDIQUI, SAMEER	
	Examiner Bradley Edelman	Art Unit 2153	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,697,942 B1	02-2004	L'Heureux et al.	713/152
	B	US-6,510,454 B1	01-2003	Walukiewicz, Miroslaw	709/206
	C	US-6,480,901 B1	11-2002	Weber et al.	709/246
	D	US-2003/0014505	01-2003	RAMBERG et al.	709/223
	E	US-2003/0172115	09-2003	MOTOYAMA, TETSURO	709/206
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Business Wire, "Embedded Web Technology to be Coupled with SNMP Agents; Networks Manageable Via the Web," Feb. 8, 1999, Business Wire, from http://www.findarticles.com/p/articles/mi_m0EIN/is_1999_Feb_8/ai_53720278, pp. 1-3.
	V	Fujitsu Siemens Computers GmbH, Emanate Master 1.3 Readme, June 2000, pp. 1-29.
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Emanate Master 1.3

Readme

Edition June 2000

Comments... Suggestions... Corrections...

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Fax forms for sending us your comments are included at the back of the manual.

There you will also find the addresses of the relevant User Documentation Department

Copyright and Trademarks

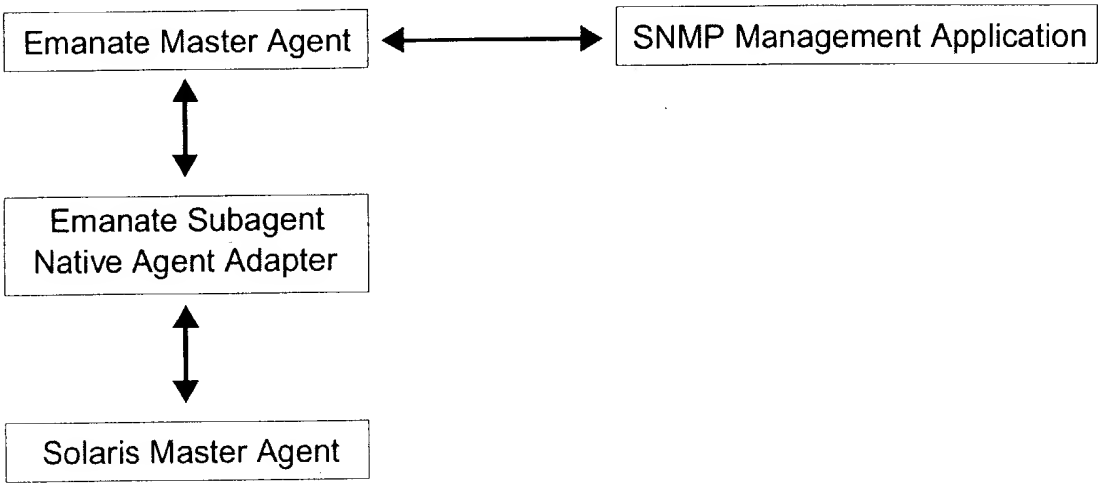
This manual was produced by
cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

This manual is printed on
paper treated with
chlorine-free bleach.

Contents

1 General

In contrast to Reliant UNIX, the Emanate Master Agent on Solaris should be viewed in combination with the Solaris Master Agent (Solaris Sun Solstice Enterprise Master Agent). Both Master Agents are connected by the Siemens Native Agent Adapter, which hangs as a Subagent on the Emanate Master Agent and passes SNMP requests to the Solaris SNMP Agent on the same machine.



The Emanate Master Agent thus takes over some of the tasks of the Solaris Master Agent, but does not replace it completely. MIB objects, which are supported by the Master Agent or one of its Subagents, will continue to be supplied exclusively by this agent. This refers, for example, to all MIB objects in the MIB2 tree under Solaris.

The Solaris Master Agent must open Port `161`, on which SNMP requests are received, for this purpose. The Emanate Master Agent takes over this port, while the Solaris Master Agent awaits SNMP requests directed to it on another port (default is `8161`). The Native Agent Adapter sets up a connection to this Solaris Master Agent port. The MIB objects, for which the Native Agent Adapter is to establish contact with the Solaris Master Agent, are defined in a configuration file.

In the standard version, this relates to all objects in mib_2 and in the sun tree:

`read 1.3.6.1.2.1` (mib_2)

`read 1.3.6.1.4.1.42` (sun)

SNMP requests that relate to MIB objects in mib_2 or in the sun tree are thus passed directly from the Emanate Master via the Native Agent Adapter to the Solaris Master Agent. The configuration file likewise defines whether read or also write access is enabled to these MIB objects. Read access is supported exclusively in the standard version.

If write access is also required to these objects, "read" must be replaced with "readwrite" (see man page `adaptagt`(1M))).

Irrespective of the type of SNMP request to the Emanate Master Agent (v1, v2* or v3), the Native Agent Adapter generates an SNMPv1 request to the Solaris Master Agent. The community string, used here is generally

- `public` as read community and
- `private` as write community.

If the Solaris Master Agent expects other communities, the Native Agent Adapter must be restarted with the appropriate values (see man page `adaptagt`(1M)).

The functionality of the Solaris Master Agent and its Subagents is preserved following connection to the Emanate Master Agent. If, for example, the mib2 Subagent rejects write access to its objects under Solaris, then it is also not possible to modify these objects via the Emanate Master Agent and the Native Agent Adapter. For information on the Solaris components, see the documentation under Solaris (e.g. man page `snmpdx`(1M) and man page `mibiisa`(1M)).

1 Converting Existing SNMP Configurations

SNMP Research Release 15.1 (SMAWsnmpm 1.3) software supports the SNMPv3 Administration Framework. Previous releases of the same software used different administration frameworks. This chapter describes the steps necessary to update an existing SNMP configuration for Release 15.1 (SMAWsnmpm 1.3).

1.1 SNMPv1 Administration Framework

SNMP Research products in and before Release 11.x) implemented the SNMPv1 Administration Framework. Starting with Release 12.1 (SMAWsnmpm 1.0), SNMP Research products which were compiled to support only the SNMPv1 protocol (and later SNMPv2c) could optionally use the SNMPv1 Administration Framework as a simple alternative to the SNMPv2 configuration.

In SNMP Research products, the SNMPv1 authentication information is only used by SNMP entities which have a command responder application and/or a notification originator application. Authorization information is only used by SNMP entities which have a command responder application. Since these kinds of applications are usually found only in "agent" entities, the configuration information in the SNMPv1 Administration Framework is located only in the *snmpd.cnf* configuration file.

When the *snmpd.cnf* file to be converted contains only SNMPv1 configuration information, there are two kinds of configuration entries which are important. These are the *community* entry and the *trap* entry.

1.1.1 Converting community Entries

An entry that begins with the *community* TAG contains a community string which can be used by "manager" entities to gain access to the MIB in the "agent" entity. The format of the VALUE clause is:

```
community-name IP-address privileges community-id
```

The fields in a *community* entry should be converted as follows:

community-name
: is the community string. First, this string should appear in the *communityName* field of a *communityEntry* entry. The *communityGroupName* field of this *communityEntry* entry must contain a valid *vacmGroupName* (READ or WRITE in the example below). See the man page snmpd.cnf(4) for more information.

  As an example, if the community-name is public, the *communityEntry* might be:

```
communityEntry localSnmpID public READ localSnmpID default - nonVolatile
```

  Next, the community string must be assigned as a principal to an access control group. The community string should appear in the *vacmSecurityName* field of a *vacmSecurityToGroupEntry* entry. The *vacmSecurityModel* field of this *vacmSecurityToGroupEntry* entry should be snmpv1 to make the string be an SNMPv1 community string, or snmpv2c to make the string be an SNMPv2c community string. To make this string be both an SNMPv1 and an SNMPv2c community string, create two *vacmSecurityToGroupEntry* entries. See the man page snmpd.cnf(4) for more information.

  Continuing with the example above, there may be two *vacmSecurityToGroupEntry* entries:

```
vacmSecurityToGroupEntry snmpv1 public READ nonVolatile
vacmSecurityToGroupEntry snmpv2c public READ nonVolatile
```

IP-address
: indicates the remote site for which this community is valid. If the IP address is 0.0.0.0, the *communityTransportLabel* field of the *communityEntry* entry should be a dash (–). If the IP address is not 0.0.0.0, this address should appear in the *snmpTargetAddrTAddress* field of an *snmpTargetAddrEntry* entry, and the value of the *communityTransportLabel* field of the *communityEntry* entry should appear in the list of (space-separated) tags in *snmpTargetAddrTagList* field of the *snmpTargetAddrEntry* entry. Additionally, the *snmpTargetAddrTDomain* field should be snmpUDP-Domain, and the *tgtAddressMask* field should be 255.255.255.255:0. See the man snmpd.cnf(4) for more information.

privileges
: is a coarse authorization identifier with an implied MIB view of "see-everything" (The SNMPv1 Administration Framework supports only a trivial access control mechanism). To convert this configuration information, first verify that a *vacmViewTreeFamilyEntry* entry similar to the following exists (create an entry if one does not already exist):

```
vacmViewTreeFamilyEntry All iso - included nonVolatile
```

This entry creates a MIB view that is arbitrarily called "All" which includes a point just below the root of the MIB tree, iso.

Next, if the *privileges* field is `READ`, verify that *vacmAccessEntry* entries similar to the following exist (create them if they do not already exist):

```
vacmAccessEntry READ - snmpv1  noAuthNoPriv exact All - - nonVolatile
vacmAccessEntry READ - snmpv2c noAuthNoPriv exact All - - nonVolatile
```

If the *privileges* field is `WRITE`, verify that *vacmAccessEntry* entries similar to the following exist (create them if they do not already exist):

```
vacmAccessEntry WRITE - snmpv1  noAuthNoPriv exact All All - nonVolatile
vacmAccessEntry WRITE - snmpv2c noAuthNoPriv exact All All - nonVolatile
```

NOTE:

- The strings `READ` and `WRITE` in the new entries are arbitrary strings.
- The correct string (either `READ` or `WRITE` in this example) must appear in the *communityGroupName* field of the corresponding *communityEntry*.
- The *vacmAccessReadViewName* field should identify the "All" MIB view.
- If the *privileges* field is `WRITE`, then the *vacmAccessWriteViewName* field should identify the "All" MIB view.
- The *vacmAccessNotifyViewName* field can identify the "All" MIB view if this community string will also be used for sending Trap messages (see the following section.

`community-id`
is unused by the SNMPv3 Administration Framework and does not need to be converted.

1.1.2 Converting trap Entries

An entry that begins with the *trap* TAG contains the IP address of an SNMP "manager" entity to which an SNMPv1 (and later SNMPv2c) Trap is sent. The format of the VALUE clause is:

```
trap-community-name IP-address
```

The fields in a trap entry should be converted as follows:

- The *trap-community-name* is the community string which is sent in the Trap message. First, this string should appear in the *snmpTargetParamsSecurityName* field of a *snmpTargetParamsEntry* entry.

  If the Trap should be sent as an SNMPv1 message, the *snmpTargetParamsMPModel* field of the *snmpTargetParamsEntry* entry should be zero (0), the *snmpTargetParamsSecurityModel* field should be `snmpv1`, and the *snmpTargetParamsSecurityLevel* field should be `noAuthNoPriv`. If the Trap should be sent as an SNMPv2c message, the *snmpTargetParamsMPModel* field of the *snmpTargetParamsEntry* entry should be one (1), the *snmpTargetParamsSecurityModel* field should be `snmpv2c`, and the *snmpTargetParamsSecurityLevel* field should be `noAuthNoPriv`.

  If the Trap should be sent as both an SNMPv1 message and as an SNMPv2c message, there should be two *snmpTargetParamsEntry* entries. As an example, if the *trap-community-name* is `test2`, the *snmpTargetParamsEntry* entry(ies) might be:

```
snmpTargetParamsEntry v1ExampleParams 0 snmpv1 test2
noAuthNoPriv \
nonVolatile

snmpTargetParamsEntry v2cExampleParams 1 snmpv2c test2
noAuthNoPriv \
nonVolatile
```

- The *IP-address* indicates the destination address of the notification; i.e., where to send the *Trap*. First, this address should appear in the *snmpTargetAddrTAddress* field of a *snmpTargetAddrEntry* entry, and the *snmpTargetAddrParams* field of the *snmpTargetAddrEntry* entry should be the same as the *snmpTargetParamsSecurityName* field of the *snmpTargetParamsEntry* entry(ies). Additionally, the *snmpTargetAddrTDomain* field of the *snmpTargetAddrEntry* entry should be `snmpUDPDomain`, the *tgtAddressMask* field should be `255.255.255.255:0`, and the *snmpTargetAddrTagList* should be a unique string, such as `ConvertedTraps`. See the man page `snmpd.cnf`(4) for more information. As an example, if the trap entry is

```
trap test2 192.147.142.254
```

  then the *snmpTargetAddrEntry* entry(ies) might be:

```
 snmpTargetAddrEntry MyMgrEntry1 snmpUDPDomain
192.147.142.254:0 100 3 \
ConvertedTraps v1ExampleParams nonVolatile 255.255.255.255:0
```

```
snmpTargetAddrEntry MyMgrEntry2 snmpUDPDomain
192.147.142.254:0 100 3 \
ConvertedTraps v2cExampleParams nonVolatile 255.255.255.255:0
```

Next, the unique string which is the value of snmpTargetAddrTagList in the snmpTargetAddrEntry entry(ies) should appear in the snmpNotifyTag field of a snmpNotifyEntry entry. The snmpNotifyTypefield of the snmpNotifyEntry entry should be "trap". Continuing with the example above, the new entry might be:

```
snmpNotifyEntry 31 Console trap nonVolatile
```

Next, verify that a vacmViewTreeFamilyEntryentry similar to the following exists (create an entry if one does not already exist):

```
vacmViewTreeFamilyEntry All iso - included nonVolatile
```

This entry creates a MIB view that is arbitrarily called "All" which includes a point just below the root of the MIB tree, iso.

Next, the *trap-community-name* must be assigned as a principal to an access control group. The community string should appear in the *vacmSecurityName*-field of a *vacmSecurityToGroupEntry* entry. The *vacmSecurityModel*field of this *vacmSecurityToGroupEntry* entry should `snmpv1` to cause an *SNMPv1* Trap messages to be sent, or `snmpv2c` to cause an *SNMPv2c* Trap messages to be sent. To cause both *SNMPv1* and *SNMPv2c* messages to be sent, create two *vacmSecurityToGroupEntry* entries. See the man page `snmpd.cnf`(4) for more information. Continuing with the example above, there may be two *vacmSecurityToGroupEntryentries*:

```
vacmSecurityToGroupEntry snmpv1  public TRAP nonVolatile
vacmSecurityToGroupEntry snmpv2c public TRAP nonVolatile
```

Finally, verify that a *vacmAccessEntry* entry exists which contains the trap-community-namein the *vacmGroupNamefield*, `snmpv1` or `snmpv2c` as desired in the *vacmAccessSecurityModel* field. The *vacmAccessNotifyViewName* field should identify the "All" MIB view. If such a *vacmAccessEntry* entry does not exist, create an entry for each type of *Trap* message (SNMPv1 or SNMPv2c) to send. For example:

```
vacmAccessEntry TRAP - snmpv1 noAuthNoPriv exact - - All nonVolatile
vacmAccessEntry TRAP - snmpv2c noAuthNoPriv exact - - All nonVolatile
```

The string `TRAP` in the new entries is an arbitrary string. If the same community string will be used to send *Traps* as well as to generate *Get*, *Set*, etc. SNMP request, one can simply set the *vacmAccessNotifyViewName* field in an existing *vacmAccessEntry* entry to the "All" MIB view.

1.2 Party-based SNMPv2 Administration Framework

SNMP Research products from Release 12.1(SMAWsnmpm 1.0) to Release 12.3 (SMAWsnmpm (SMAWsnmpm 1.0) implemented the *Party*-based SNMPv2 Administration Framework. Today, both the protocol and security administration framework of Party-based SNMPv2 have moved to the Historic state and is no longer supported.

In the Party-based SNMPv2 Administration Framework, a *party* was an encapsulation of the security information necessary to perform authentication and privacy operations. In the SNMPv3 Administration Framework, a *user* serves the same purpose. However, the configuration information for parties can not be easily converted into configuration information for users. This section describes how to understand SNMPv2 parties well enough to dermine the IP address(es) of the "manager" entities which may have sent requests to or received *Traps* from the agent.

In a bilingual SNMP agent of the period, a special kind of party, called an *rfc1157Domain party*, contained an SNMPv1 community string instead of SNMPv2 authentication and privacy information. This section also describes how to understand rfc1157Domain parties well enough that the same community strings can be configured in the SNMPv3 Administration Framework.

1.2.1 Configuration Files

SNMP Research entities which implemented the Party-based SNMPv2 Administration Framework did not store the SNMP configuration information in the *snmpd.cnf* file. Instead, these entities used configuration files with names ending with the extension `.pty`. Also, the format and usage of the *mgr.cnf* file was radically different than in Release 15.1 (SMAWsnmpm 1.3).

The discussion of the Party-based SNMPv2 Administration Framework in this section will be limited to the following file:

```
agt.pty
```

The following files should be discarded:

```
acl.pty
context.pty
mgr.cnf
```

```
mgr.pty
view.pty
```

1.2.2 Deciphering Network Address Information

This section describes how to determine the IP address of the SNMP "manager" and "agent" entities from the configuration information for SNMPv2 parties in the *agt.pty* file. Each entry in the *agt.pty* file consists of seven lines:

```
PartyName PartyDiscriminator
TDomain TAddress Port Lifetime MaxMsgSize
partyIndex partyStorageType partyLocal partyAuthClock
AuthPublicSecret
AuthPrivateSecret
PrivPublicSecret
PrivPrivateSecret
```

The following fields contain useful information:

`TAddress`

This field contains an IP address for the party, but the purpose of the address depends upon the *TDomain* and *partyLocal* fields:

- When *TDomain* is `rfc1157Domain`, this field, in conjunction with the *Port* field, defines a *Trap* destination address. This field is also used for source address authentication for SNMPv1 requests.
- When *TDomain* is `snmpUDPDomain` and *partyLocal* is `false`, this field contains the IP address of a remote SNMP entity. Usually, this remote entity is the SNMP "manager" which contains the command generator application and/or notification receiver application.
- When *TDomain* is `snmpUDPDomain` and *partyLocal* is `true`, this field contains the IP address of the local SNMP entity (the "agent").

`partyLocal`

This field is meaningless unless the *TDomain* is `snmpUDPDomain`. For *snmpUDPDomain*, the values are:

- `true`: the IP address in *TAddress* is for the 'local' host
- `false`: the IP address in *TAdress* is not the 'local' host, but it is the address of a remote SNMP entity.

An example party table entry is shown here:

```
initialPartyId.192.147.142.16.2 3
snmpUDPDomain 192.147.142.16 162 300 1458
2 nonVolatile false 0
-
74 68 69 73 74 68 69 73 74 68 69 73 74 68 69 34
-
-
```

In the example entry above:

- The *TDomain* is `snmpUDPDomain`, indicating that this is a SNMPv2 party.
- The *TAddress* is `192.147.142.16`, indicating that an SNMP entity existed on the machine whose IP address is `192.147.142.16`.
- Since *partyLocal* is `false` and it is a SNMPv2 party, the *TAddress* field contains the IP address of a remote SNMP entity, probably the SNMP "manager" which contains the command generator application and/or notification receiver application.

1.2.3 Deciphering Community String Information

This section describes how to locate SNMPv1 community strings within the configuration information for SNMPv2 parties in the *agt.pty* file. Each entry in the *agt.pty* file consists of seven lines:

```
PartyName PartyDiscriminator
TDomain TAddress Port Lifetime MaxMsgSize
partyIndex partyStorageType partyLocal partyAuthClock
AuthPublicSecret
AuthPrivateSecret
PrivPublicSecret
PrivPrivateSecret
```

The following fields contain useful information:

`TDomain`
: This field will contain *rfc1157Domain* to indicate that this is an SNMPv1 community entry in the party table.

`AuthPrivateSecret`
: When *TDomain* is `rfc1157Domain`, this field contains community string stored as a series of hexadecimal numbers. Each number is the ASCII value for the corresponding character in the community string. For example, the entry would be `70 75 62 6c 69 63` for the community string `public`.

An example party table entry is shown here:

```
initialPartyId.192.147.142.16.31 1
rfc1157Domain 192.147.142.16 162 300 1458
31 nonVolatile true 0
-
70 75 62 6c 69 63
-
-
```

In the example entry above:

- The *TDomain* is `rfc1157Domain`, indicating that this is an SNMPv1 community entry in the party table.
- The *TAddress* and *Port* fields indicate that traps should be sent to port `162` at IP address `192.147.142.16`. This field also indicates that SNMPv1 packets will only be accepted if they come from `192.147.142.16`.

  The *AuthPrivateSecret* contains the community name: `0 75 62 6c 69 63`,which decodes to `public`.

1.3 SNMPv2* Administration Framework

SNMP Research products from Release 14.1 to Release 14.2 (SMAWsnmpm 1.1 and 1.2) implemented the SNMPv2* Administration Framework. This framework, like its successor the SNMPv3 Administration Framework, encapsulates the security information necessary to perform authentication and privacy operations into the familiar concept of "users" and "passwords." This similarity allows configuration information from the SNMPv2* Administration Framework to be easily converted into configuration information for the SNMPv3 Administration Framework.

With Release 15.1 (SMAWsnmpm 1.3) software products, SNMP Research provides a program which automatically converts configuration files from the earlier releases implementing the SNMPv2* Administration Framework. The following section describes this conversion program and its use.

The cfgcvt Configuration Conversion Program

The *cfgcvt* configuration conversion program reads an SNMP Research SNMP configuration file (ending with extension `.cnf`) as input and creates a new configuration file (ending with extension `.new`) for use by SNMP entities which implement the SNMPv3 Administration Framework.

The new configuration file must be manually renamed with the .cnf extension before it can be used by SNMP Research software.

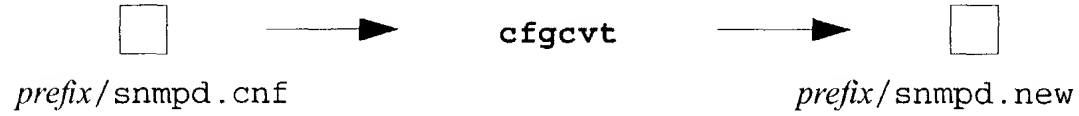


Figure 5: The default behavior of cfgcvt

Defaults

The *cfgcvt* assumes that the input file to convert is called `snmpd.cnf`, the output file to create is called `snmpd.new`, and that the filename prefix (directory path name) is `/etc/srconf/agt` on UNIX systems. Figure 1 shows the default behavior.

Command Line Arguments

The behavior of the *cfgcvt* program is modified by the following command line options.

`-agt`

Converts the configuration file used by SNMP Research SNMP "agent" entities. This is the default.

`-mgr`

Converts the configuration file used by SNMP Research SNMP "manager" entities. When the *-mgr* command-line is used with *cfgcvt*, the default input file name changes to `mgr.cnf`, the default output file name changes to `mgr.new`, and the default prefix changes to `_`.

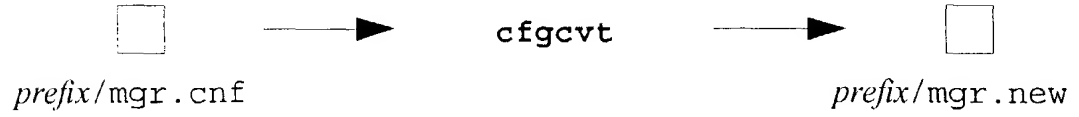


Figure 6: The behavior of cfgcvt with the -mgr option

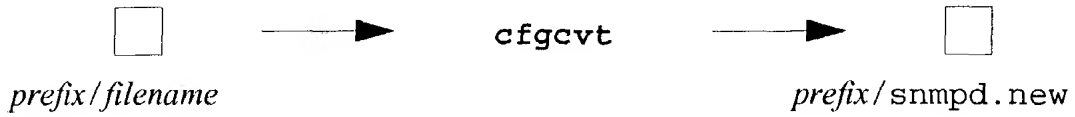


Figure 7: The behavior of cfgcvt with the -in option

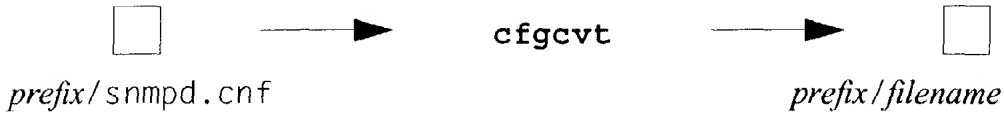


Figure 8: The behavior of cfgcvt with the -out option

The filename prefix (directory path name) is `/etc/srconf/mgr` on UNIX systems. Figure 2 depicts the modified behavior.

`-in` *filename*
: Changes the name of the input file to filename. Figure 3 depicts the modified behavior.

`-out` *filename*
: Changes the name of the output file to filename. Figure 4 depicts the modified behavior.

`-prefix` *directory*
: Changes the prefix to directory. This command line argument overrides any environment variables which may be set.

Environment Variables

The behavior of the *cfgcvt* program is modified by the following environment variables.

`SR_MGR_CONF_DIR`
: This environment variable changes the prefix to the input file when the *-mgr* command-line argument is being used.

`SR_AGT_CONF_DIR`
: This environment variable changes the prefix to the input file when the defaults are being used or when the *-agt* command-line argument is being used.

Note that the value of these environment variables may be overridden with the *-prefix directory* command-line argument.

Using cfgcvt on UNIX Systems

Before running *cfgcvt* on UNIX Systems, perform the following steps.

1. Become root. The program must be started by superuser.

   ```
   % su
   ```

2. Specify the location of the mgr.cnfconfiguration file:

   ```
   # setenv SR_MGR_CONF_DIR /etc/srconf/mgr
   ```

To start *cfgcvt*, type the program's name followed by any desired command-line arguments.

```
% cfgcvt
```

Understanding Warning Messages

This section explains the warning messages which may be generated by *cfgcvt*.

```
WARNING:*****************************************************
WARNING: There are v2ContextEntry lines in the input file.
WARNING: These cannot be converted to the output file.
WARNING:*****************************************************
```

The *cfgcvt* program prints the above message if it finds one or more *v2ContextEntry tags* in the input file. The definition of context information differs between the SNMPv2* Administration Framework and the SNMPv3 Administration Framework and can not be converted.

```
WARNING:*****************************************************
WARNING: There are acEntry lines in the input file which
WARNING: contain non-empty acContextNameMask values.
WARNING: These cannot be converted to the output file.
WARNING: You may wish to hand-edit the resulting output
WARNING: file and change some of the vacmAccessEntry lines
WARNING: to use a 'prefix' value for vacmAccessContextMatch.
WARNING: *****************************************************
```

The *cfgcvt* program prints the above message if it finds one or more *acContextNameMask* fields in the input file which contain a value other than a dash (–). The definition of context name 'wildcards' differs between the SNMPv2* Administration Framework and the SNMPv3 Administration Framework and requires human intervention to be converted.

```
WARNING: ****************************************************
WARNING: There are notifyEntry lines in the input file.
WARNING: If multiple entries contain the same
WARNING: notifyIdentityName and notifyContextName
WARNING: values, but different notifyViewName values,
WARNING: the extra notifyViewName values will not be used
WARNING: in the output file.
WARNING: ****************************************************
```

The cfgcvt program prints the above message if it finds one or more *notifyEntry TAG*s in the input file. The *notifyEntry* entries can be automatically converted, but if more than one *notifyEntry* entry exists with the same identity and context but different MIB views, only the first MIB view is retained in the output file.

```
BackupFile: Warning, cannot backup config file.
            at line 1899 in file scanfile.c
WriteConfigFile: Warning, cannot backup config file.
                 at line 1736 in file scanfile.c
```

The *cfgcvt* attempts to make a backup copy of converted configuration before starting the conversion process. The first time *cfgcvt* is executed for a particular directory, a converted file does not already exist, so the above message is printed. This message could also mean that the output file from a previous conversion exists, but it can not be overwritten.

```
OpenConfigFile: can't open /etc/srconf/agt/snmpd.cnf with mode 1
                at line 207 in file k_fileio.c
```

The *cfgcvt* program does not have sufficent permission to open the configuration file for reading, or the file does not exist in the named directory.

```
OpenConfigFile: can't open /etc/srconf/agt/snmpd.new with mode 2
                at line 207 in file k_fileio.c
```

The *cfgcvt* program does not have sufficent permission to open the output file for writing.

```
WriteConfigFile: WARNING, CANNOT RESTORE OLD CONFIG FILE
                 at line 1850 in file scanfile.c
```

This message means that the directory containing the original configuration file is not writable by *cfgcvt*. The original file is never modified by *cfgcvt*, so no configuration information from the original file is lost.

Fujitsu Siemens Computers GmbH
User Documentation
81730 Munich
Germany

Fax: (++49) 700 / 372 00000

email: manuals@fujitsu-siemens.com
http://manuals.mchp.siemens.de

Comments
Suggestions
Corrections

Submitted by

Comments on Produktname Version
Handbuchtitel

Bestellnummer

Embedded Web Technology to be Coupled With SNMP Agents; Networks Manageable Via the Web

Business Wire, Feb 8, 1999

MAYNARD, Mass.--(BUSINESS WIRE)--Feb. 8, 1999--

Agranat Systems and SNMP Research International Integrate Technologies

Accelerating Development of Web-based SNMP Management Solutions

Agranat Systems, Inc., the market leader in embedded Web server technology, today announced an agreement to integrate EmWeb/NM with SNMP Research International's Master Agent EMANATE(R). By coupling the technologies, a Web browser can access SNMP-based management information directly through EMANATE. The alliance permits original equipment manufacturers to quickly and easily develop Web-based interfaces that access SNMP data without writing additional code.

Because EmWeb/NM delivers the full power of simple network management protocol (SNMP) management to the Web by leveraging existing management information bases (MIBs), developers are spared months of re-coding. EmWeb/NM's fully compliant HTTP 1.1 server ensures that any authorized browser can access a networking product from anywhere on the network. Joined at inception, EmWeb/NM with EMANATE accelerates Web-based network management solution development.

"The integration of EMANATE with EmWeb/NM provides a revolutionary means for manufacturers to capitalize on the management applications of the Web," said Jeff Case, founder and CTO of SNMP Research International. "For the first time datacom equipment manufacturers have the industry leading tools to create efficiently managed networks via the Web. SNMP Research is pleased to partner with a vendor of Agranat's integrity and technological leadership."

Datacom and network equipment manufacturers now have the tools to create solutions that configure, manage and monitor systems using a familiar Web interface. EmWeb/NM employs a variety of unique development techniques to accelerate the creation of dynamic Web pages without requiring CGI-based applications. Providing numerous management choices, EmWeb/NM is compatible to a variety of network systems. Industry leaders and innovators recognize that EmWeb/NM's C compiled embedded server provides the smallest, most efficient implementation available.

"The integration of the technologies will further decrease datacom OEM's time-to-market for their Web managed products" said Ian Agranat, president and CEO of Agranat Systems. "As we look forward to a long-standing relationship with SNMP Research, Agranat is committed to supporting all future versions of EMANATE and will

About SNMP Research International:

SNMP Research International offers network management solutions, by supplying SNMP software products including the following: the EMANATE Extensible Agent system; security configuration and customization tools for HP OpenView; a Packaged Agent System designed to ease the OEM's task of adding manageability to embedded systems; intelligent agent Mid-Level Managers; and specialized MIB implementations of IETF standards including the SysAppl MIB, Host Resources MIB, RMON, UPS, and others.

In addition, the company's new Legacy Adapter to Internet (LATIN) class of products provides an affordable solution to managing legacy systems under an SNMP framework. LATIN translates legacy system messages and alarms to SNMP format without requiring custom-built MIBs or agents.

Interoperability, simplicity, robustness, and portability are hallmarks of SNMP Research International products. The company has achieved significant worldwide market penetration with hundreds of multi-national, long-standing customers shipping thousands of SNMP-capable products and services based on SNMP Research code. Founded in 1988, SNMP Research is a well-established leader in developing and supporting standards-based management technology.

For more information about SNMP Research's products and services, contact them by phone at their headquarters in Knoxville TN at 423-579-3311, or by e-mail at info@int.snmp.com. You may also access their Web site at www.int.snmp.com.

About Agranat Systems, Inc.

Agranat Systems Inc. is a software development company specializing in embedded Web technologies for datacomm, telecomm, imaging and other manufacturing industries seeking to provide added value and competitive advantages for their products and customers by implementing Web-based management capabilities. Organizations that have licensed EmWeb technology include Hewlett-Packard, Bay Networks, 3Com, Nortel, Compaq, Lucent, Nokia, Cabletron and many other companies worldwide. Agranat Systems is a member of the World Wide Web consortium (W3C) and the Desktop Management Task Force (DMTF) and actively participates on the Internet Engineering Task Force (IETF) for Web standards development. To learn more about Agranat and EmWeb, please visit Agranat's web site: http://www.agranat.com. You may also call corporate headquarters at 1 (978) 461-0888, the Mountain View, CA sales office at 1 (650) 903-2233, or send E-mail to sales@agranat.com.